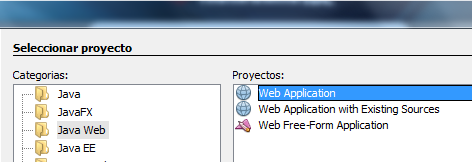
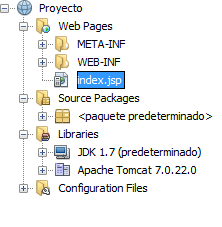
Servlets.

El primer paso es crear nuestro proyecto en el cual esta incluido el servlets, seria el siguiente

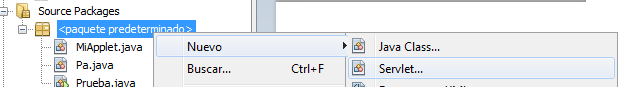


Nuevo proyecto y seleccionamos Java Web->Web Application. Lo siguiente es solo asignar un nombre para nuestro trabajo y se crea el proyecto con la siguiente contenido.

Donde tenemos un index.jsp el cual será el principal dentro de nuestra pagina dinámica.

Para poder crear un servlets damos clic secundario en source packages y seleccionamos un nuevo servlets

Definimos un nombre para el servlets.



El contenido incial del jsp y del servlet es el siguiente



Como sabes el método post no envía información en este caso es un “nombre” tipo texto con un etiqueta llamada nombre y un tamaño de 45, el cual es enviado directamente a nuestro servlet con el nombre de PA

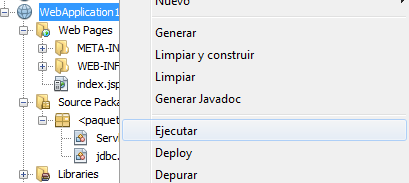
El código que tenemos en el servlet es el capturar la información enviada desde el jsp así que el siguiente código que tenemos es el siguiente

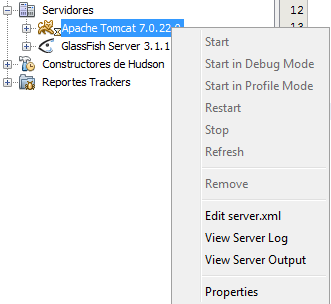
Los servlet a diferencia de los jsp tienen la ventaja de poder trabajar con lenguaje html y de forma natural con java.

Para ser un servlet, una clase debería extender [HttpServlet](http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpServlet.html) y sobreescribir doGet o doPost (o ambos), dependiendo de si los datos están siendo enviados mediante GET o POST. Estos métodos toman dos argumentos: un [HttpServletRequest](http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpServletRequest.html) y un [HttpServletResponse](http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpServletResponse.html).

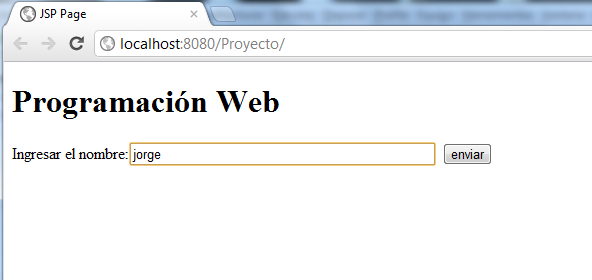
El HttpServletRequest tiene métodos que nos permiten encontrar información entrante como datos de un FORM, cabeceras de peticiòn HTTP, etc. El HttpServletResponse tiene métodos que nos permiten especificar líneas de respuesta HTTP (200, 404, etc.), cabeceras de respuesta (Content-Type, Set-Cookie, etc.), y, todavía más importante, nos permiten obtener un [PrintWriter](http://java.sun.com/products/jdk/1.2/docs/api/java/io/PrintWriter.html) usado para envíar la salida de vuelta al cliente. Para servlets sencillos, la mayoría del esfuerzo se gasta en sentencias println que generan la página deseada. Observamos que doGet y doPost lanzan dos excepciones, por eso es necesario incluirlas en la declaración. También observamos que tenemos que importar las clases de los paquetes java.io (para PrintWriter, etc.), javax.servlet (para HttpServlet, etc.), y javax.servlet.http (para HttpServletRequest y HttpServletResponse). Finalmente, observamos que doGet y doPost son llamados por el método service, y algunas veces queremos sobreescribir directamente el método service, por ejemplo, para un servlet que maneje tanto peticiones GET como POST.

Ahora bien para ejecutar y poder visualizar nuestro trabajo necesitamos las siguientes consideraciones, tener inicializado el servidor que utilicemos, si trabajamos con base de datos que tenga la conexión y los drives correctos. Solo resta postrarnos en nuestro proyecto clic secundario y seleccionamos la opción ejecutar

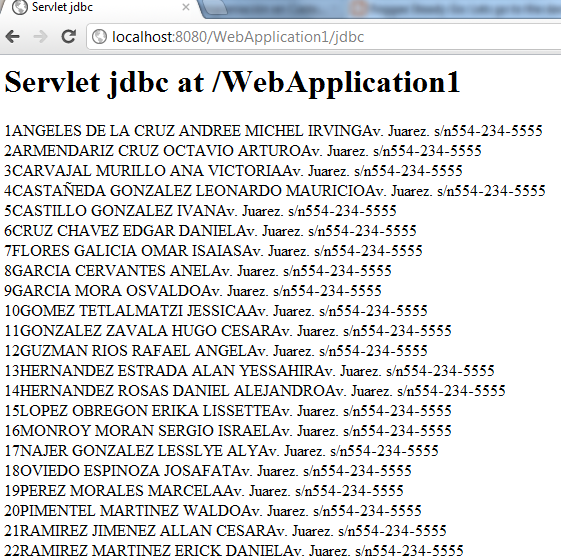




Lo que visualizamos en nuestro navegador es lo siguiente



Aquí se encuentra el estilo en html y editamos en nuestro sjp

aquí muestra como resultado lo que tenemos programado nuestro servlet en mi caso yo lo programe para acceder a una base de datos y extraer la información .

Mi código en servlet es el siguiente

public class jdbc extends HttpServlet {

private Connection con;

private Statement set;

private ResultSet rs;

String URL = "jdbc:mysql://localhost:3306/sams";

String userName = "root";

String password = "root";

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

Class.forName("com.mysql.jdbc.Driver");

con = DriverManager.getConnection(URL,userName,password);

set = con.createStatement();

System.out.println("Se ha conectado");

} catch (Exception e) {

System.out.println("No se ha conectado");

}

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet jdbc</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet jdbc at " + request.getContextPath() + "</h1>");

out.println("</body>");

out.println("</html>");

try {

String idsocio,nombre,direccion,tel;

set = con.createStatement();

rs = set.executeQuery("SELECT \* FROM socio");

while (rs.next()) {

idsocio = rs.getString("idsocio");

nombre = rs.getString("nombre");

direccion = rs.getString("direccion");

tel = rs.getString("tel");

out.println("<tr><td>" + idsocio + "</td><td>" + nombre + "</td></tr>"+ direccion + "</td></tr>"+ tel + "</td></tr>");

out.println("<br>");

}

rs.close();

set.close();

} catch (Exception e) {

System.out.println("No lee de la tabla");

}

} finally {

out.close();

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP

\* <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP

\* <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}